



MIPI DSI näytön implementointi STM32 mikrokontrollerille

Eetu Piukkula
Ohjaaja: Jukka Lahti

Piukkula E. (2019) MIPI DSI näytön implementointi STM32 mikrokontrollerille. Oulun yliopisto, Elektroniikan ja tietotekniikan tutkinto-ohjelma.

Tiivistelmä

Tässä kandidaatintyössä tarkastellaan kuinka MIPI (Mobile Industry Processor Interface) DSI (Display Serial Interface) näyttöä voidaan ohjata STMicroelectronics:n ARM-pohjaisella (Acorn RISC Machine) mikrokontrollerilla. Työssä tarkastellaan MIPI DSI standardin tuomia hyötyjä ja vaatimuksia sekä fyysisestä että ohjelmisto-teknisestä näkökulmasta katsottuna sovellettuna STM32F7 mikro-ohjaimelle.

Avainsanat: MIPI, DSI, STM32F7, näyttö, LCD-paneeli, kandidaatintyö, mikrokontrolleri, mikroprosessori, ARM

Piukkula E. (2019) Implementation of MIPI DSI display to STM32 microcontroller. University of Oulu, Degree Program in Electronics and Communication Engineering.

Abstract

Topic of this Bachelor's Thesis is how to control MIPI (Mobile Industry Processor Interface) DSI (Display Serial Interface) display with ARM (Acorn RISC Machine) microcontroller manufactured by STMicroelectronics. Topics discussed include pros and cons of MIPI DSI from hardware and software point of views.

Key words: MIPI, DSI, STM32F7, display, LCD, B.Sc. degree, microcontroller, ARM

Sisällysluettelo

Tiivistelmä	2
Abstract	3
Sisällysluettelo	4
Johdanto	5
1. MIPI DSI	6
1.1. MIPI Alliance ja DSI	6
1.2. Command Mode ja Video Mode	6
1.3. DSI-väylän differentiaalisuus	7
1.4 DSI väylän liikenne	8
1.5 DSI pakettien muoto	8
1.5 DSI komennot ja kirjoittaminen	11
1.6 Näyttöpaneelin vaatimukset STM32 prosessorille	12
2. Piirilevyn toteutus	13
2.1 Fyysisen toteutuksen suunnittelu ja toteutus	13
3. Ohjelmiston toteutus	15
3.1 Ohjelmiston suunnittelu ja konfigurointi	15
3.2 Ohjelmiston logiikka ja toteutus	16
4. Pohdinta	18
5. Yhteenveto	18
Lähdeluettelo	19

Johdanto

Oulun yliopiston Formula Student Oulu ry:n harrasteprojektista alkunsa saanut idea, kehittää formula-autoon mittaristo, jossa visualisoitaisiin auton tietoja kuljettajalle. Tämä tapahtuisi graafisen näyttöpaneelin sekä vaihtovalojen avulla. Löysimme yhteistyökumppanin, joka tarjosi sponsorisopimuksella käyttöömme tehtävään erinomaisesti sopivan LCD-paneelin. Kyseinen paneeli oli avonaisen rata-auton ohjaamoon täydellisesti sopiva Ortustech:n valmistaman 4.8” 720p LCD-paneeli, jossa rajapintana ohjaimena käytettävään prosessoriin suuntaan olisi MIPI DSI. Paneelin erikoiseksi tekee se, että rakenteesta johtuen paneeli pystyy käyttämään osan ambienttivalosta pikselin valaisuun. Toisin sanoen huomattavasti matalammalla taustavalon teholla saavutetaan kirkkaissa olosuhteissa todella hyvä luettavuus. Tämä tekee paneelista todella hyvän avonaisen ohjaamon omaavaan formula-autoon.

Heräsi kysymys, kuinka harrastelija voisi käyttää normaalisti mobiiliteollisuuden käyttämää standardia omassa projektissaan. Tutkittuamme asiaa tiimimme päätyi käyttämään STM32F769-sarjan mikrokontrolleria, sillä kyseisen prosessori sarjan prosessorit olivat markkinoilla lähes ainoita mikrokontrollereita, joissa on tuki MIPI DSI rajapinnalle. STM32F7 prosessoriperheen sydän on yksiytiminen ARM Cortex-M7-ydin, joka toimii maksimissaan 216 MHz kellotaajuudella. STM32F7 prosessoreita on saatavilla eri konfiguraatioissa sekä paketoinneissa. DSI rajapinta oli tarjolla muutamassa eri versiossa. Päädyimme käyttämään silloin tehokkainta versioita. Paketoinniksi valitsimme LQFP-176 (Low-profile Quad Flat Package), sillä halusimme paketoinnin, jossa jalat ovat näkyvissä ja mitattavissa kehittämistyön aikana. Lisäksi halusimme valmistusteknisistä syistä olla käyttämättä BGA-piirejä (Ball Grid Array).

Tässä työssä selvitetään kehittyneen harrastajan mahdollisuutta käyttää MIPI DSI väylää omassa projektissaan. Työn aikana suunniteltiin ja rakennettiin Formula Student Oulu ry:lle osana elektroniikka tiimiä.

1. MIPI DSI

1.1. MIPI Alliance ja DSI

MIPI Alliance on useiden mobiiliteknologia jättien vuonna 2003 perustama liittouma, jonka pohjimmaisena tarkoituksena on luoda standardeja mobiilitekniikan alalle. Tavoitteena on luoda sovelluksesta riippumatta yleisesti hyväksytyjä standardeja, jotka helpottavat ja optimoivat mobiililaitteiden suunnittelua ja rakentamista eri toimijoiden välillä. MIPI Alliancella on useita osa-alueita, jotka ovat luoneet useita yhteisiä standardeja. Yksi näistä on DSI, eli sarjamuotoinen data-rajapinta näytön tarvitsemalle datalle. Muita standardeja ovat esimerkiksi CSI (Camera Serial Interface) ja SPMI (System Power Management Interface). [1]

DSI eli Display Serial Interface on yksi MIPI Alliancen kehittämistä standardeista. DSI:n suunnittelussa pyrittiin luomaan yleisesti hyväksytty, halpa ja etenkin pienen fyysisen jalanjäljen omaava rajapinta näytön ja prosessorin välille. DSI väylä koostuu differentiaalisista korkeataajuisista signaaleista muodostuvista väyläpareista. DSI väylässä on aina yksi kellolinja-pari sekä ainakin yksi datalinja-pari. Datalinjoja voi olla useita, riippuen näytön vaatimasta kaistasta. Yksi datalinja voi maksimissaan tarjota kellotaajuudesta riippuvan siirtonopeuden. Näin ollen datalinja-parien lukumäärä kasvattamalla on mahdollista kasvattaa väylässä siirrettävän datan maksimi määrää lineaarisesti. [1]

Esimerkiksi käyttämämme STM32F7 mikro-ohjaimen kelloväylän nopeus on maksimissaan 500 MHz ja datalinja-pareja on kaksi kappaletta. Yksi datalinja-pari kykenee näin ollen siis 500Mb/s siirtonopeuteen. Molempien kaistojen yhteen laskettu nopeus on siis 1Gb/s. Tämä riittää hyvin käyttämällemme 720 x 1280 resoluution näytölle jonka väriavaruus on kolme tavua per pikseli sekä virkistystaajuus 30Hz. [2, 3, 4]

1.2. Command Mode ja Video Mode

DSI näyttöjä on pääpiirteissään kahdenlaisia, CM (Command Mode) ja VM (Video Mode). CM näyttöissä on oma, sisäinen volatiili muisti vähintään yhtä kuvapuskuria (eng. framebuffer) varten. Tällainen paneeli on kalliimpi kuin VM paneeli, sillä näytön sisäinen nopea muisti on melko kallista. Tämä voi olla sekä hyvä että huono asia, riippuen onko paneelia ohjaavalla prosessorilla omaa muistia. Monessa mobiiliteollisuuden käyttämässä prosessorissa, varsinkin SoC-suorittimissa (System on a Chip) on usein paljon omaa nopeaa muistia. Tällöin paneelin sisäisen muistin hyödyt laskevat ja kalliimpi hinta nostaa kustannuksia turhaan. CM:n hyväksi puoliksi voi listata matalamman kaistan käytön DSI väylässä, sillä väylässä liikkuva data on vain mitä vaaditaan sisäisen muistin päivittämiseen sekä tähän liittyvän päivitysnopeuden, sillä kuvapuskuri on suoraan näytössä. [1]

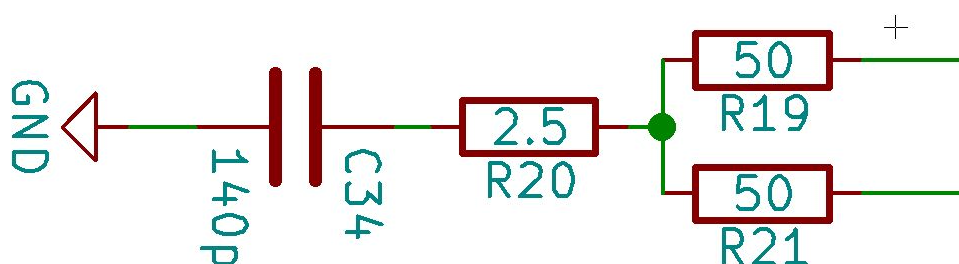
Toinen yleinen tyyppi on VM-paneeli, jossa ei ole omaa muistia. FSO:n mittaristossa käytetty paneeli on tällainen. VM-paneelissa kuvapuskuri sijaitsee ohjaavalla prosessorilla taikka tähän yhdistetyllä nopealla volatiilisella ulkoisella muistilla.

VM-paneelin hyöty on suhteellinen halpuus verrattuna CM-paneeliin, sillä paneelissa ei ole kallista muistia. VM-paneelin huono puoli on se, että paneeli vaatii jatkuvaa data-virtaa DSI väylän kautta. Tämä datavirta tulee ohjaimen omasta sisäisestä tai ulkoisesta muistista. Jos muisti taikka prosessori ei kykene luomaan tarvittavaa datavirtaa johtaa se graafisiin virheisiin näytöllä. VM-paneelissa DSI väylässä oleva datamäärä on korkea. Korkean datamäärän takia on suunnitteluvaiheessa otettava huomioon riittävä kaistanleveys DSI väylässä suhteessa näytön resoluutioon, väriavaruuteen sekä näytön virkistystaajuuteen. [1]

1.3. DSI-väylän differentiaalisuus

DSI on sarjamuotoinen differentiaalinen väylä, joka tarkoittaa, että väylän ominaisuuksilla on sekä hyviä että huonoja puolia. Suurin haaste DSI väylässä on sen suuri nopeus, maksimissaan 500 MHz. Suuri taajuus pakottaa suunnittelussa olemaan tarkkana sovituksen kanssa. Signaalin heijastuminen ja vaihe on otettava huomioon piirilevyn fyysisessä suunnittelussa. Nopeiden signaalien kuparoitteen pituuksien tulisi olla mahdollisimman lähellä toistensa pituuksia sekä samaan aikaan mahdollisimman lyhyitä. Sarjamuotoisuus mahdollistaa datan tiiviin pakkaamiseen suhteessa pinta-alaan. Verrattuna rinnakkaiseen näyttöpaneeliin vastaavilla suoritusarvoilla, olisi rinnakkaisessa paneelissa noin 20 eri signaalia verrattuna tässä tapauksessa DSI:n kuuteen, neljä data-signaalia ja kaksi kello-signaalia. Nämä ovat D0P, D0M, D1P, D1M, CLKP ja CLKM (Data 0+, Data 0-, Data 1+, Data 1-, Kello + ja Kello -). Pienempi pinta-ala fyysisessä toteutuksessa on todella arvokas varsinkin mobiiliteollisuudessa, jossa vuosien edetessä laitteen sisällä on jatkuvasti vähemmän ja vähemmän tilaa. [1]

DSI väylä on differentiaalinen. Tämä tarkoittaa sitä, että sekä datalinjat että kellolinjat koostuu johdinpareista. Molemmissa osissa johdinparia kulkee sama signaali, mutta päinvastaisella logiikalla. Tämä mahdollistaa signaalille hyvän kuuluvuuden myös häiriöisessä ympäristössä sillä häiriö näkyy molemmissa johtimissa ideaalitilanteessa samanlaisena ja nollaantuu signaalia käsiteltäessä. Väylän differentiaalinen luonne helpottaa fyysistä toteutusta korkeataajuisessa väylässä. Väylän differentiaalisuuden takia tulee väylä olla terminoitu. Terminointitavaksi valittiin niinsanottu "T-terminointi".



Kuva 1. Kuva valitusta differentiaalisen väylän terminoinnista. Kuva KiCadista.

1.4 DSI väylän liikenne

DSI väylässä tapahtuvaa liikennettä on kahdenlaista. Itse kuva ja siihen liittyvä tieto liikkuu aina HS-moodissa (High-speed mode). HS liikennettä tapahtuu kaikilla käytössä olevilla linjoilla. DSI speksin mukaan maksimi tiedonsiirtonopeus on 1.5Gb/s, mutta STM32F7 tukee maksimissaan vain kolmannesta tästä eli 500Mb/s per linja. HS-liikenne on aina yksisuuntaista prosessorilta näytölle. [1]

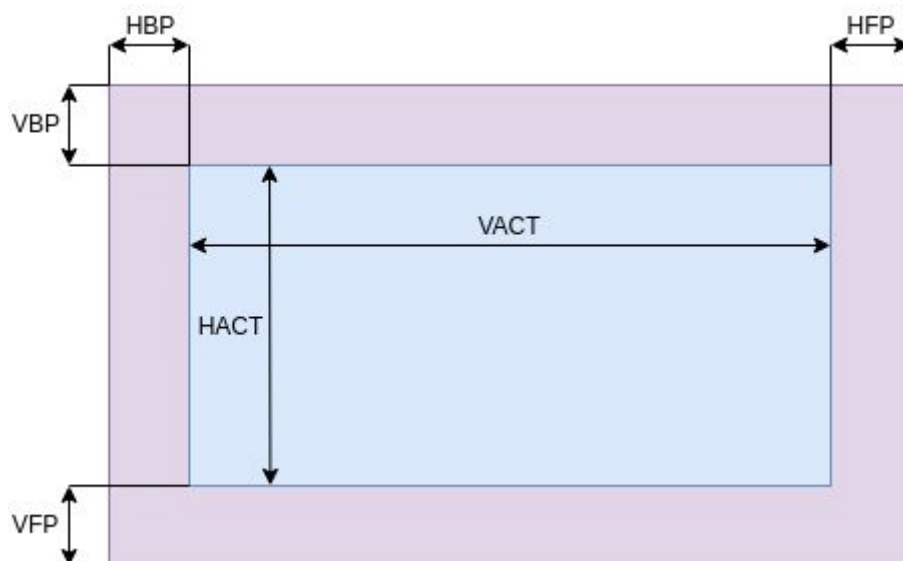
Toinen liikenne on LP moodissa (Low-power mode) kulkevaa konfigurointiin liittyvää liikennettä. LP moodissa maksimi nopeus on 10Mb/s ja sitä tapahtuu vain data-linjassa nolla ja se voi olla molemminsuuntaista. LP moodia käytetään paneelin käynnistyessä ensimmäistä kertaa konfiguroimaan paneelin pienen volatiiliin kontrollerin muistiin halutut parametrit. DSI protokolla mahdollistaa paneelin uudelleen alustamisessa taikka konfiguroimisen kesken videon lähettämisen sekä LP että HS moodissa. Mittaristossa oleva paneeli tukee tätä mahdollisuutta. Todellisuudessa tilanne, jossa paneeli pitäisi uudelleen konfiguroida ei tapahdu juuri koskaan. [1]

HS- ja LP-signaalien ero on selkeästi nähtävissä mikäli linjaa tutkii oskilloskoopilla. LP-moden liikenne tapahtuu sekä hitaammin että myöskin suuremmilla logiikan jännitetasoilla. LP-moodin logiikan yläraja on tyypillisesti 1.2V ja alaraja noin 0V (0V +/- 50mV). Sen sijaan HS-moodissa signaalin logiikkatasot ovat suhteessa yhteiseen bias-jännitteeseen. Bias-jännitteen ollessa 200mV logiikan yläraja on 270mV ja alaraja 130mV. Todellisuudessa oikeassa sovelluksessa signaalien poikkeama bias-jännitteestä on yleensä noin 100mV. [1]

1.5 DSI pakettien muoto

DSI paketti koostuu useista osista. Paketin osasten määrä ja arvot määräytyy prosessorin ominaisuuksien ja suoritustehon perusteella ja osa näyttöpaneelin ominaisuuksista.

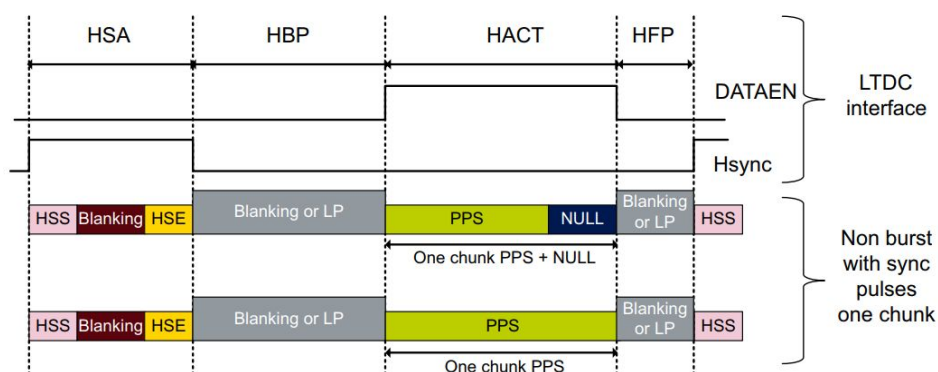
Näyttöpaneelin mukaan määräytyvät ominaisuudet ovat ajoitukseen liittyviä parametrejä. Itse kuvan lisäksi näytölle lähetetään sekä korkeussuunnassa että leveyssuunnassa ajoituksen tarkentavia parametreja. Nämä parametrit on määritetty näyttöpaneelin valmistajan tarjoamassa taulukossa ja kuvataan yleensä joko pikseleinä taikka riveinä suhteessa piirtotasoon. Paketti koostuu neljästä osasta. Järjestyksessä HSA (H-Sync Active), HBP/VBP (Horizontal/Vertical Back Porch), HACT (H-Active) ja HFP/VFP (Horizontal/Vertical Front Porch). HSA kuvastaa paketin alkua, HBP/VBP ja HFP/HFP kuvastavat virtualisoitua aluetta piirtoalueen ulkopuolella, jota on sekä ennen että jälkeen piirtoalueen. HACT kuvastaa dataa, eli piirtoalueen dataa. [1]



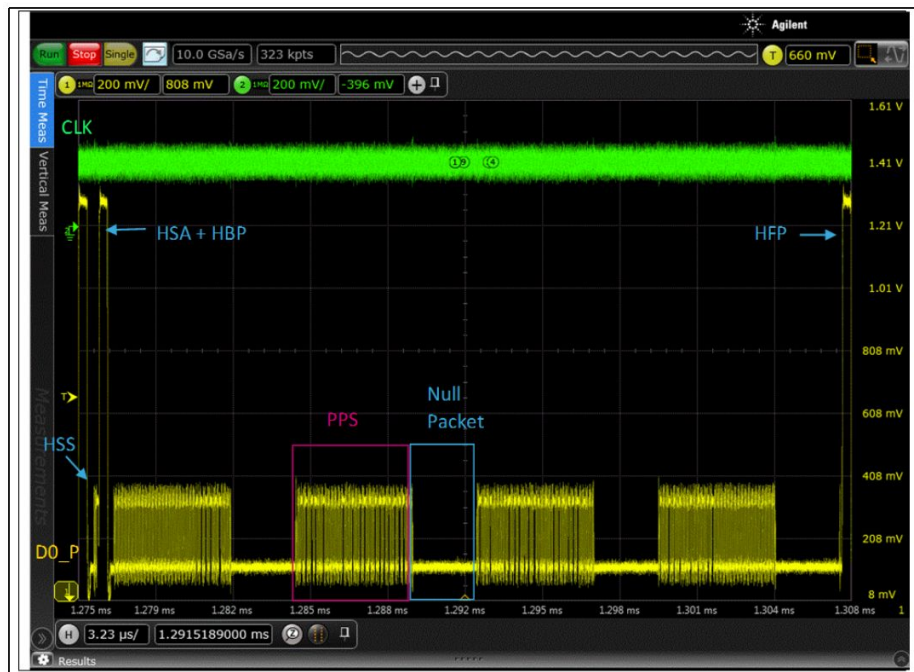
Kuva 2. Havainnollistava kuva näyttöpaneelin ajoituksesta

Item	Symbol	Condition	Rating			Unit	Remark
			MIN	TYP	MAX		
VSYNC Frequency	f _{VSYNC}	Fig. C	54	58.1	66	Hz	D0P,D0N , D1P,D1N , D2P,D2N , D3P,D3N , CP,CN
VSYNC Pulse Width	VS		3	3	-	line	
Vertical Back Porch	VBP		3	3	-	line	
Vertical Front Porch	VFP		6	6	-	line	
Vertical Display Period	VDISP	Fig. C	-	1280	-	line	
HSYNC Pulse Width	HS		-	15	-	clk	
Horizontal Back Porch	HBP		-	70	-	clk	
Horizontal Front Porch	HFP		-	91	-	clk	
Horizontal Display Period	HDISP	Fig. C	-	720	-	clk	
Minimum HS time	t _{HS}		0.2	-	-	us	
Minimum HBP time	t _{HBP}		1	-	-	us	
Minimum HFP time	t _{HFP}		1	-	-	us	
Minimum HS + FHP + HBP time	t _(HS+HBP+HFP)		2.2	-	-	us	

Kuva 3. Valitun näyttöpaneelin valmistajan ilmoittamat ajoitusparametrit



Kuva 4. Kaaviokuva DSI-väylässä tapahtuvasta liikenteestä. Kuva STM



Kuva 5. Esimerkki oskilloskoopilla tutkitusta DSI-väylästä. Kuva STM

Proessori sisältää sisään rakennetun näyttöajurin LTDC (LCD-TFT Display Controller), joka on vastuussa kuvapuskuriin kirjoitettavan datan raameista. Raameissa määrätään valitun paneelin mukaan vaihtuvat parametrit sekä kuvapuskurin muistialueen osoite. STM32 prosessoreissa LTDC voi toimia kahdella tapaa. LTDC voi toimia itsenäisesti rinnakkais-muotoisesti käyttäen esimerkiksi Motorola 6800-protokollaa. Tällöin LTDC vastaa sekä ohjelmistollisesta että fyysisestä toteutuksesta. LTDC voi myös toimia väliprosessina, jolloin se vastaa vain ohjelmistollisesta osasta. Tällöin LTDC-ohjain ohjaa luomansa paketit prosessorin sisäisesti DSI-ohjaimelle, joka vastaa fyysisestä rajapinnasta muokaten paketit uudelleen DSI-väylään ja näyttöpaneelille sopiviksi. [1]

Jotta LTDC-ohjain sekä DSI-ohjain toimivat oikein yhdessä on tärkeää sovittaa LTDC:n tuottama tietovirta yhtä suureksi kuin mitä DSI kykenee sitä käyttämään. Mikäli tämä yhtälö ei ole tasapainossa johtaa se datan hukkumiseen ja toimimattomaan kokonaisuuteen. Yleensä tämä tasapaino ei ole täydellinen, jolloin tilannetta kompensoidaan lisäämällä HACT alueen loppuun tyhjää dataa (NULL). Tämä toimii vain jos DSI on suhteessa nopeampi kuin LTDC, jolloin DSI-bussiin jää käyttämätöntä kapasiteettia. [1]

$$LTDCPixelClock = (VSA + VBP + VACT + VFP) \times (HSA + HBP + HACT + HFP) \times FrameRate$$

$$LTDCPixelClock = (3 + 3 + 1280 + 6) \times (15 + 70 + 720 + 91) \times 26 = 30\,098\,432 \text{ Hz (30 MHz)}$$

$$\frac{VACT \times BytePerPixel + 12 \times NullSize}{DSILaneByteClock \times NumberOfLanes} = \frac{DSILaneByteClock}{LTDCPixelClock}$$

$$\frac{720 \times 3 + 12 \times NullSize}{62.5 \times 2} = \frac{62.5}{30} \rightarrow NullSize = 828$$

1.5 DSI komennot ja kirjoittaminen

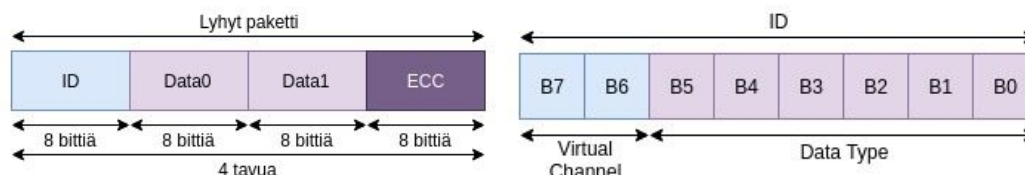
VM DSI paneelia ohjataan standardisoiduilla käskyrakenteilla. Yleisimmät käskyt ovat kuvan synkronointiin liittyvät (0x01, 0x11, 0x21 ja 0x31) sekä käynnistäessä suoritettavaan konfigurointiin liittyvät DCS kirjoituskäskyt (0x05, 0x15 ja 0x39). STM32 mikrokontrollerin valmistajan tarjoama HAL (Hardware Abstraction Layer) kirjastojen avulla tapahtuva kuvan piirtäminen ja siihen liittyvät komennot tapahtuvat automaattisesti. Käynnistyessä tehtävä näyttöpaneelin konfigurointi on tehtävä manuaalisesti. Prosessi on helppo ja HAL-kirjaston tarjoamat komennot on helppo suorittaa. [1, 4]

Data Type Hex	Description	Size
01 h	Sync Event , V Sync Start	Short
11 h	Sync Event , V Sync End	Short
21 h	Sync Event , H Sync Start	Short
31 h	Sync Event , H Sync End	Short
22 h	Shut Down Peripheral Command	Short
32 h	Turn On Peripheral Command	Short
08 h	End of Transmission Packet (EoTp)	Short
05 h	DCS WRITE , no parameters	Short
15 h	DCS WRITE , 1 parameters	Short
06 h	DCS READ , no parameters	Short
37 h	Set Maximum Return Packet Size	Short
09 h	Null Packet , no data	Long
19 h	Blanking Packet , no data	Long
39 h	DCS Long Write/write_LUT Command Packet	Long
0E h	Packed Pixel Stream , 16-bit RGB , 5-6-5 Format	Long
1E h	Packed Pixel Stream , 18-bit RGB , 6-6-6 Format	Long
2E h	Loosely Packed Pixel Stream , 18-bit RGB , 6-6-6 Format	Long
3E h	Packed Pixel Stream , 24-bit RGB , 8-8-8 Format	Long

Kuva 6. Prosessorilähtöiset datapaketti tyypit. Kuva näyttöpaneelin datasheetistä.

Prossessorilta näyttöpaneelille kirjoittaminen hoituu kolmella eri DCS (Display Command Set) komennolla, “DCS kirjoitus lyhyt - parametreilla”, “DCS kirjoitus lyhyt - ilman parametrejä” sekä “DCS kirjoitus - pitkä”. DCS kirjoituskomentojen paketit ovat toistensa kanssa hyvin samanlaisia. [1]

Lyhyissä paketeissa on neljä yhden tavun pituista osaa, ID, Data0, Data1 ja ECC. Järjestyksessä ID osa koostuu kuuden bitin pituisesta komennon datatyyppiistä sekä sitä seuraavan kahden bitin pituisesta näyttöpaneelin fyysisestä osoitteesta. Paneelin osoite on pysyvä valmistajan määrittämä osoite, joka lukee valmistajan tarjoamassa dokumentissa. [1]



Kuva 7. Kuvaus lyhyestä DSI paketista ja ID osan rakenteesta

Parametrit 0x05 ja 0x15 ovat nimensä puolesta harhaan johtavia. Komento 0x05 eli “DCS kirjoitus lyhyt - ilman parametrejä” todellisuudessa sisältää parametrin Data0, muttei parametria Data1. Vastaavasti komento 0x15 eli “DCS kirjoitus lyhyt - yksi parametri” sisältää kaksi tavua kuormaa, Data0 ja Data1. Oikeaa komentoa miettiessä voi ajatella, että komennossa on aina vähintään yksi tavu hyötydataa, jolloin komennossa lukeva parametrien lukumäärä lasketaan ensimmäisen datatavun lisäksi. [1]

Pitkä kirjoituspaketti koostuu kolmesta osasta, PH (Packet Header), PD (Packet Data) sekä PF (Packet Footer). Erona pitkän ja lyhyen paketin välillä on syötettävän datan määrä. PH vastaa lähetettävän datan muodosta ja koosta. PH:n ensimmäinen tavu on samankaltainen kuin lyhyen kirjoituksen ID osa, jossa kerrotaan laitteen ID sekä käytettävän paketin nimi. Seuraavana PH sisältää kaksi tavua tietoa siitä kuinka paljon tietoa on odotettavissa PD osassa. Viimeinen tavu PH:ssä sisältää ECC (Error-Correcting Code) tarkistussumman. PD sisältää datan, joka halutaan siirtää. Se voi olla mikä tahansa tavumäärä, jonka voi ilmoittaa 16 bitin etumerkittömällä luvulla, joka ilmoitettiin PF:ssä. Viimeisenä tulee PF, jossa DSI-ohjain laskee 16 bittiä pitkän tarkistussumman PD:stä, varmistaakseen viestin oikeellisuuden. [1]



Kuva 8. Kuvaus pitkästä DCS paketista

1.6 Näyttöpaneelin vaatimukset STM32 prosessorille

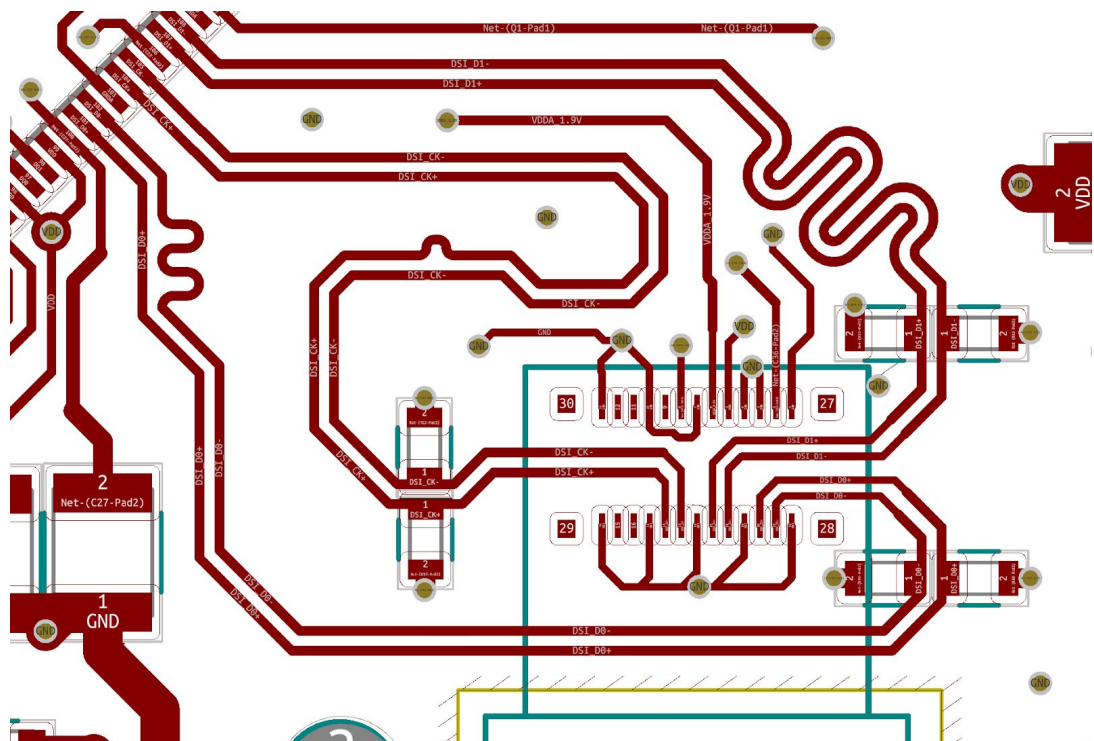
STM32F7 prosessoreissa on sisäistä SRAM muistia 512 kt, muttei läheskään tarpeeksi kuvapuskuriksi, joka valitsemallamme näytölle on noin 2.76Mt ($720 \times 1280 \times 3$, näytön resoluutio kerrottuna jokaisen pikselin väriavaruudella). Olisi hyvä jos kuvapuskureita olisi kaksi kokonaista kappaletta, jotta voidaan suorittaa kaksinkertaista puskurointia. Kaksikertaisessa puskuroinnissa takana olevaa kuvapuskuria päivitetään kun edessä olevaa kuvapuskuria piirretään näytölle. Kun piirtäminen on valmis, siirretään koko takana oleva kuvapuskuri edessä olevaan. Näin saadaan aikaiseksi parempi luettavuus tiedon vaihtuessa näytöllä. Haittapuolena kaksinkertainen puskurointi vaatii enemmän suoritustehoa sekä muistilta että prosessorilta. [2,3]

STM32F7 prosessoreissa on FMC (Flexible Memory Controller, prosessorivalmistajan rajapinta volatiileille muisteille), jonka avulla prosessoriin on helppo liittää ulkoista RAM muistia. Ulkoiseksi muistiksi valittiin Alliance Semiconductorin valmistama AS4C1M16S SDRAM piiri. Ulkoinen muisti on 16Mb kokoinen. Databussin leveys on maksimi mitä STM32F769 tukee, eli 16 bittiä. Osoitteen leveys on 11 bittiä. [2, 3]

2. Piirilevyn toteutus

2.1 Fyysisen toteutuksen suunnittelu ja toteutus

Piirilevyn fyysinen suunnittelu on suoraviivaista. Sekä DSI-väylä että ulkoisen muistin vaatimat nopeat ja haastavat väylät tulee ottaa huomioon ja pyrkiä tekemään ne ilman kompromisseja. Näyttöpaneelin liittimen sekä prosessorin jalkojen välitys ovat samaan kokoluokkaa. Valittiin suurin mahdollinen johtimen leveys (0.254 mm) sekä suurin mahdollinen johtimien välinen etäisyys (0.254 mm), jotka sopivat molempien päiden padeille. Prosessorin sijoittelulla varmistettiin, että matka prosessorilta näyttöpaneelin liittimelle on mahdollisimman lyhyt. Fyysisen sijoittelun takia osa differentiaalisista linjoista ovat suhteessa eripituisia. KiCadin differentiaalityökaluilla varmistettiin, että jokainen linja on suhteessa yhtä pitkä toisiin pareihin sekä differentiaaliseseen pariinsa nähden. Tarkkuudeksi valittiin $\pm 1\text{mm}$. SaturnPCB-työkalulla tarkistettuna linjan differentiaalinen impedanssi on $97.25\ \Omega$, sekä impedanssi maata vasten on $56.65\ \Omega$. Linjalla olevat sovitus komponentit sijoitettiin siten, että toinen pääty komponentista on mahdollisimman paljon linjan kuparoinnin päällä. Näin minimoidaan heijastukset ja epäideaalisuudet. Ensimmäiset vastaukset sijoitettiin samalle puolelle kuin linja itse on, mutta sovituksen kondensaattori sekä yhteinen vastus sijoitettiin pienen tilan takia levyn toiselle puolelle mahdollisimman lähelle läpivientä.



Kuva 9. DSI väylän fyysinen toteutus piirilevyssä. Kuva ruutukaappaus KiCadista.

Differential Pairs
 Conductor Width (W) **0.254 mm**
 Conductor Spacing (S) **0.254 mm**
 Conductor Height (H) **0.2 mm**
 W/H = 1.270
 S/H = 1.270

Target Zdiff **100 Ohms**

Formula Restrictions:
 0.1 < W/H < 3.0
 0.1 < S/H < 3.0

Zdiff **97.225 Ohms**
Zo **56.646 Ohms**
 +/- Tolerance = 10%

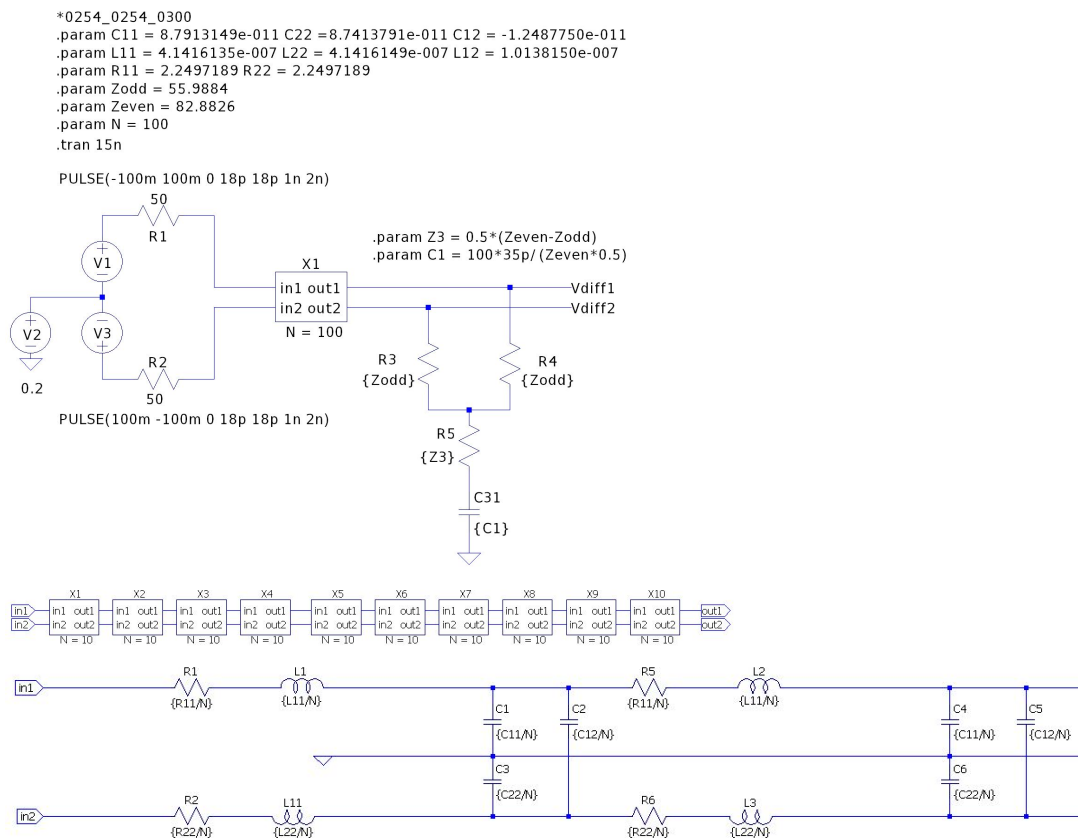
Options
 Base Copper Weight
☐ 9um
☒ 18um
☐ 35um
☐ 53um
☐ 70um
☐ 88um
☐ 106um
☐ 145um
☐ 178um
 Plating Thickness
☐ Bare PCB
☒ 18um
☐ 35um
☐ 53um
☐ 70um
☐ 88um
☐ 106um
 Differential Layer
☒ Edge Cpld Ext
☐ Edge Cpld Int Sym
☐ Edge Cpld Int Asym
☐ Edge Cpld Embed
☐ Broad Cpld Shld
☐ Broad Cpld NShld
 Units
☐ Imperial
☒ Metric
 Substrate Options
 Material Selection
Custom
 Er **4.7** Tg (°C) **130**
 Temp Rise (°C) **20**
 Temp in (°F) = 36.0
 Ambient Temp (°C) **22**
 Temp in (°F) = 71.6
 Print Solve!

Information
 Total Copper Thickness 36 um
 Via Thermal Resistance N/A
 Via Count: 10
 Conductor Temperature N/A
 Temp in (°C) = N/A
 Temp in (°F) = N/A
 Via Voltage Drop N/A

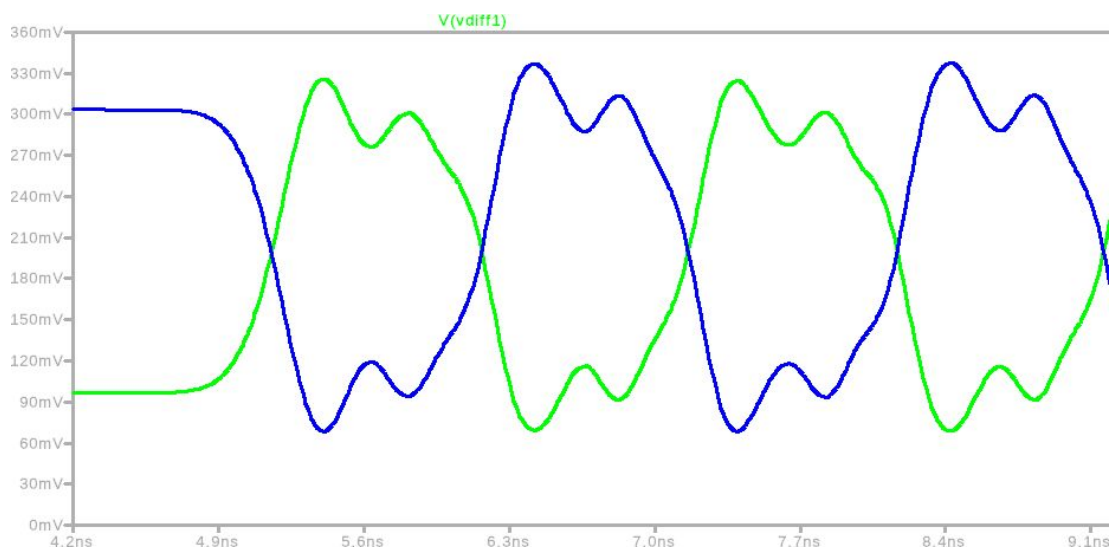
SATURN PCB DESIGN, INC.
 Turnkey Electronic Engineering Solutions

Kuva 10. Ruutukaappaus SaturnPCB työkalusta, jossa esitettynä impedanssi toteutetun DSI-väylän arvoilla.

Differentiaalisessa väylässä päätettiin käyttää T-terminointia. T-terminoinnin vastusten arvot saadaan simuloimalla malli LTSpicessä. LTSpice mallia varten käytettiin TNT-työkalua, jolla luotiin fyysistä toteutusta vastaava malli. Mallia simuloimalla varmistettiin DSI väylän teoreettinen toimiminen.



Kuva 11. Ruutukaappaus LTSpice:llä tehdystä DSI-väylän simuloinnista sekä TNT-työkalulla luodusta fyysisen toteutuksen mallista. Ylin kuva kuvaa kokonaisuutta. Keskimmäinen kuva kuvaa X1-lohkoa, joka pitää sisällään kymmenen kappaletta TNT-työkalulla luotua mallia. Alimmassa kuvassa esitetään kaksi kymmenestä lohkoista



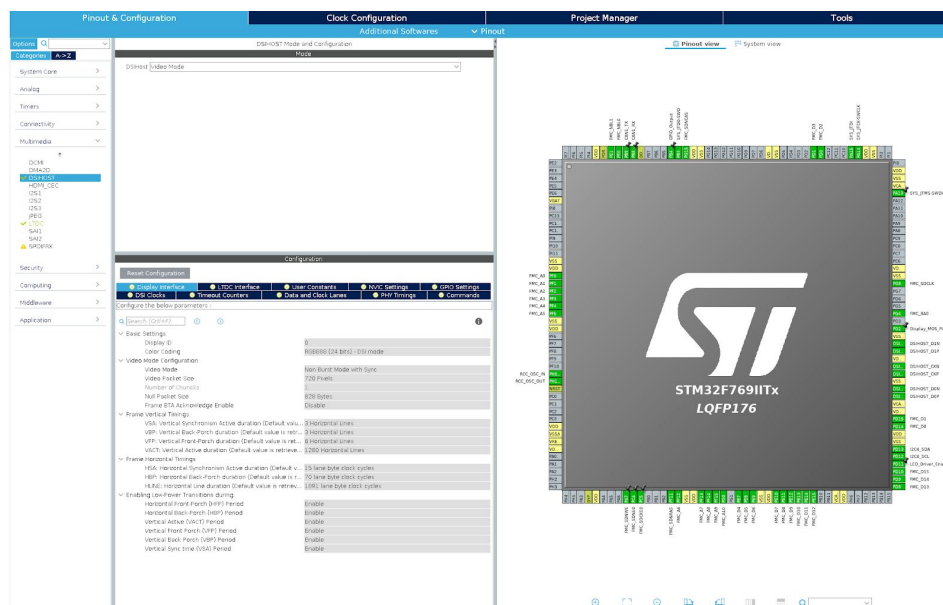
Kuva 12. Kuvakaappaus LTSpicellä suoritetusta 500 MHz kellovälän simuloinnista.

3. Ohjelmiston toteutus

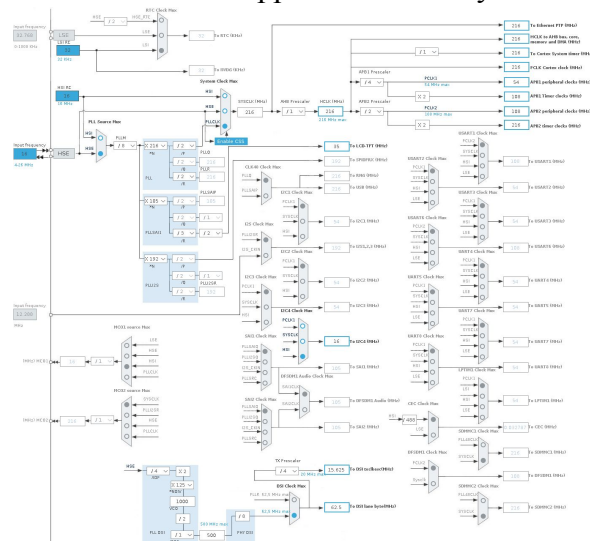
3.1 Ohjelmiston suunnittelu ja konfigurointi

Ohjelmiston kehittämisessä käytettäviä työkaluja valittaessa on varmintä sekä todennäköisimmin helpointa käyttää valmistajan omia työkaluja. Ohjelmiston konfigurointi on helppoa käytettäessä STMicroelectronics:n tarjoamaa CubeMX-ohjelmistoa. CubeMX on graafinen työkalu, jolla on mahdollista luoda projektirunko kaikille valmistajan prosessoreille sekä konfiguroida kaikkia valitun laitteen ohjaimia, kuten I2C, SPI, LTDC ja DSI. CubeMX sisältää myöskin hyvän konfigurointityökalun prosessorin sisäisille kelloille sekä sisäänrakennetun ratkaisijan kellon konfliktitilanteille. Työkalu mahdollistaa yhdellä napsautuksella luotavan projektirungon sisältäen kaikki ohjainten tarvittavat alustukset sekä näiden alustavat parametrit. Projektipohja tuo mukanaan prosessorille sopivat HAL kirjastot, joilla voi tehokkaasti ohjata haluttuja prosessorin laitteita. [2, 3, 5]

Kun prosessorille on alustettu itse prosessorin välttämättömät parametrit, kuten ulkoisen kellokiteen käyttö, voidaan työkaluun syöttää arvot, jotka on aikaisemmin laskettu ja valittu. Ensin valitaan LTDC toimivaksi DSI-tilassa. Seuraavaksi valitaan DSI toimivaksi VM tilassa. Seuraavaksi täytetään LTDC:n valitun näyttöpaneelin parametrit sekä säädetään kello-ratkaisijaan LTDC:n kellotaajuus edellä lasketuksi. Sama tehdään DSI ohjaimen valikossa, jossa kentiin lisätään aikaisemmin lasketut parametrit sekä näyttöpaneelin vaatimat arvot. Lisäksi varmistetaan että DSI:n kellon arvot ovat oikeat. Lisäksi valitaan FMC toimivaksi oikeilla parametreilla. FMC tarjoaa muistialueen näytölle. Painamalla "Generate Code" luo CubeMX projektin, jonka voi avata haluamassaan kehitysympäristössä. STMicroelectronics tarjoaa suositusta Eclipsestä muunnellun kehitysympäristön, (IDE, Integrated Development Environment) STM System Workbench. Käytettäessä valmistajan omia työkaluja pyritään minimoimaan ongelmatilanteet yhteensopivuudessa.[1]



Kuva 13. Ruutukaappaus CubeMX työkalun DSI-valikosta

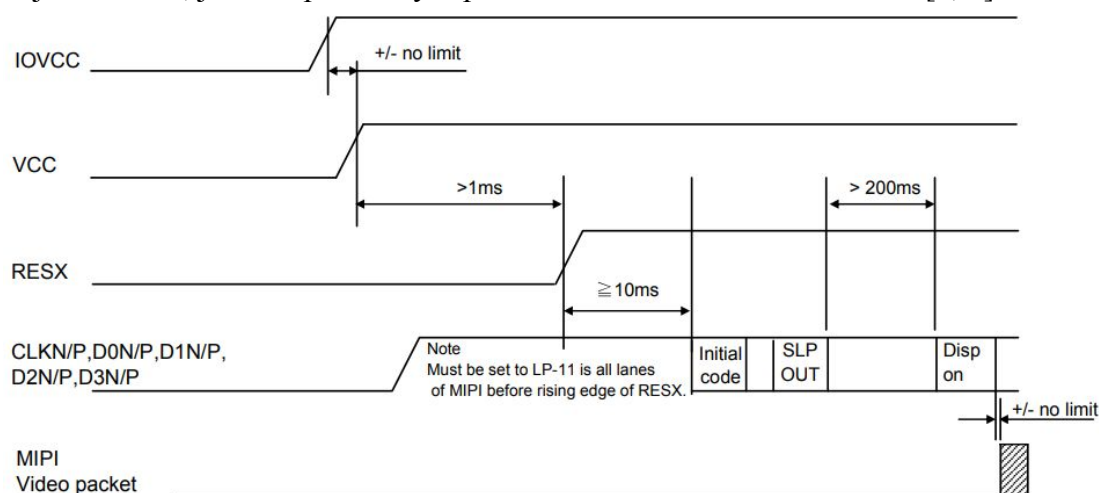


Kuva 14. Ruutukaappaus CubeMX työkalun kello-ratkaisijasta.

3.2 Ohjelmiston logiikka ja toteutus

Ohjelmisto on rakennettu sulautetulle järjestelmille tyypillisellä topografialla. Ensin ohjelmassa suoritetaan prosessorin alustaminen, jonka jälkeen alustetaan prosessoriin yhdistetyt laitteet, kuten ulkoiset ohjaimet. Seuraavaksi siirrytään itse funktionaaliseen ohjelmaan, joka elää ikuisessa “While 1” toistossa. Yksinkertaistettuna ohjelma hakee jokaisella toistolla ensin uudet arvot CAN-väylästä, jonka jälkeen ohjelma suorittaa analyysin saaduille arvoille. Seuraavana ohjelma suorittaa tiedon esittämisen käyttäjälle, jonka jälkeen aloittaa alusta tiedon hakemisella. Yksi esitystavoista on tiedon esittäminen näyttöpaneelilla.

Alustusvaiheessa ensin suoritetaan HAL-kirjastojen lataaminen, sekä prosessorin ajureiden alustaminen CubeMX:n luomissa funktioissa. Ajureihin kuuluu esimerkiksi FMC, LTDC ja MIPI. Kun laitteiden ajurit on alustettu, alustetaan laitteet itse. Näyttöpaneelin kannalta kriittisiä laitteita ovat näyttöpaneeli itse, sekä ulkoinen SDRAM muisti, jolla kuvapuskuri elää. Näyttöpaneelin tapauksessa suoritetaan 24 askelta pitkä “Power-ON Sequence”, jonka aikana DSI DCS-kirjoitus komennoilla konfiguroidaan näyttöpaneelissa oleva pieni mikrokontrolleri valmistajan tarjoamilla parametreilla. Konfigurointi alkaa LP-11 (Low Power 11, tila jossa molemmat puolet ovat “idle” tilassa) tilasta ja suorittamalla RESX (Reset) signaalille logiikan vaihtelun ylhäältä alas ja takaisin ylös. Tämän jälkeen suoritetaan 23 askelta pitkä kirjoitus vaihe, jonka lopussa näyttöpaneeli toistaa framebufferin dataa. [4, 5]



Kuva 15. Ruutukaappaus valmistajan tarjoamasta ajoitustiedosta liittyen “Power-ON Sequence:een”

Ennen kuin näyttöpaneelia käynnistetään on suotavaa suorittaa kuvapuskurin alustaminen kirjoittamalla muistialue kertaalleen, jotta muistissa jo oleva data ei päätyisi näkymään näytöllä. Mikäli kyseessä on ensimmäinen käynnistys vähään aikaan, on muisti täynnä sattumanvaraisia bittejä (valkoista melua), taikka jos suoritetaan nopea uudelleenkäynnistys voisi muistissa olla jäänteitä edellisestä tilasta, jossa muisti oli ennen uudelleenkäynnistystä.

Viimeinen asia ennen “While 1” silmukkaa on alustaa keskeytysvektori LTDC ohjaimelle. Keskeytys asetetaan laukeamaan jokainen kerta kun LTDC piirtää 1280:n rivin. Kun keskeytys laukeaa, siirtyy ohjelma funktioon, jossa suoritetaan kuvapuskurin uudelleen kirjoittaminen. Keskeytysfunktion jälkeen ohjelma palautuu edelliseen kohtaan koodissa, jossa ohjelma oli ennen kuin keskeytys laukesi. Mikäli keskeytystä ei suoritettaisi ja kuvapuskurin muistialueen uudelleenkirjoittaminen tapahtuisi sattumanvaraisessa välissä, johtaisi se todella rumiin artefakteihin, vilkkumiseen sekä vaikeasti luettavaan informaatioon näytöllä.

4. Pohdinta

Tarkoituksena oli selvittää harrastelijan mahdollisuutta käyttää mobiiliteollisuudessa yleisesti käytettyä tehokasta ja pienen fyysisen jalanjäljen omaavaa sarjamuotoista DSI-väylää, sekä tähän yhdistettyä näyttöpaneelia. Markkinoilla olevilla, jokaisen harrastajan ostettavissa olevilla sekä hyvin dokumentoiduilla tuotteilla on mahdollista luoda projekti, jossa työssä käytettyä näyttöpaneelia vastaavaa näyttöpaneelia on mahdollista käyttää. On kuitenkin huomioitava sekä näyttöpaneelin että mikrokontrollerin tekniset rajoitukset, jotka voivat estää halutun komponentin käytön. Työssä yhdistettiin sekä ohjelmistollista, että sähkötekniistä osaamista, jonka vuoksi yksi suurimmista haasteista työssä oli tutkia ongelmaa molemmilta kanteilta ja löytää paras ratkaisu molemmista näkökulmista katsottuna. Tutkielma on hyvä läpiluotaus siitä mitä tulisi ottaa huomioon vastaavaa projektia suunnitellessa ja toteuttaessa. Tulevaisuudessa STMicroelectronics on luultavasti julkaisemassa myöskin uusia STM32H7 sarjan mikrokontrollereita, jotka perustuvat samaan Cortex-M7 arkkitehtuuriin, mutta jotka toimivat noin tuplaten nopeammalla taajuudella. Näissä uusissa H7-sarjan mikrokontrollerissa saattaa löytyä nopeampi ja leveämpi DSI väylä, jolla työssä käytettävää näyttöä olisi mahdollista operoida tuplaten suuremmalla virkistystaajuudella sekä nopeammalla muistilla.

5. Yhteenveto

Formula Student Oulu ry:n konseptista alkunsa saanut projekti selvittää kuinka valistuneen harrastelijan on mahdollista käyttää tehokasta MIPI DSI-väylää omissa projekteissaan. Työssä tutkittiin DSI-väylän vaatimuksia ja ominaisuuksia etuineen ja haasteineen. Työssä seurattiin valmistajan tarjoamia resursseja, sekä yleisiä hyviä elektroniikkasuunnittelun käytäntöjä. Projekti saatiin päätökseen ja todettiin, että DSI-väylän käyttäminen omissa projekteissa on mahdollista.

Lähdeluettelo

[1] STMicroelectronics (2017), Application note AN4860. URL:
https://www.st.com/content/ccc/resource/technical/document/application_note/group0/1d/b8/33/4f/dc/0a/45/52/DM00287601/files/DM00287601.pdf/jcr:content/translations/en.DM00287601.pdf, luettu 20.02.2019

Mikrokontrollerin valmistajan tarjoama selostus DSI-protokollasta heidän tuotteillaan

[2] STMicroelectronics (2018), Reference manual RM0410. URL:
https://www.st.com/content/ccc/resource/technical/document/reference_manual/group0/96/8b/0d/ec/16/22/43/71/DM00224583/files/DM00224583.pdf/jcr:content/translations/en.DM00224583.pdf, luettu 20.02.2019

Mikrokontrollerin valmistajan tarjoama selostus mikrokontrollerin ominaisuuksista

[3] STMicroelectronics (2017), Datasheet for STM32F765xx, STM32F767xx, STM32F768Ax, STM32F769xx. URL:
<https://www.st.com/content/ccc/resource/technical/document/datasheet/group3/c5/37/9c/1d/a6/09/4e/1a/DM00273119/files/DM00273119.pdf/jcr:content/translations/en.DM00273119.pdf>, luettu 20.02.2019

Mikrokontrollerin valmistajan tarjoama selostus mikrokontrollerin ominaisuuksista

[4] Ortustech (2017), Specification for Blanview TFT-LCD Monitor COM48H4N22ULC. URL:
<http://ultran.ru/sites/default/files/catalog/svetodiody/brend/datasheets/com48h4n22ulc.pdf>, luettu 20.02.2019

Näyttöpaneelin valmistajan tarjoama selostus näyttöpaneelin ominaisuuksista

[5] STMicroelectronics (2017), Description of STM32F7 HAL and Low-layer drivers, user manual UM1905. URL:
https://www.st.com/content/ccc/resource/technical/document/user_manual/45/27/9c/32/76/57/48/b9/DM00189702.pdf/files/DM00189702.pdf/jcr:content/translations/en.DM00189702.pdf, luettu, 13.03.2019

Mikrokontrollerin valmistajan tarjoama selostus HAL-kirjastosta